

WHAT IS CLAIMED IS:

1. In a computing environment, a method comprising,  
receiving a function call via an interface, the function  
call comprising markup language data; and

5 interpreting the markup language data to cause data in a  
scene graph to be modified.

2. The method of claim 1 wherein causing data in the  
scene graph to be modified comprises causing initialization of  
10 a new instance of a visual class.

3. The method of claim 2 wherein causing data in the  
scene graph to be modified comprises invoking code to associate  
a transform with a visual object in the scene graph.

15

4. The method of claim 1 wherein causing data in a scene  
graph data structure to be modified comprises invoking code to  
place a drawing visual into the scene graph.

20 5. The method of claim 4 further comprising, causing a  
drawing context to be returned, the drawing context providing a  
mechanism for rendering into the drawing visual.

6. The method of claim 2 wherein causing data in the scene graph to be modified comprises invoking code to associate brush data with a visual object in the scene graph.

5        7. The method of claim 6 wherein the brush data comprises receiving data corresponding to a solid color.

8. The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a linear gradient  
10 brush and a stop collection comprising at least one stop.

9. The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a radial gradient brush.

15

10. The method of claim 6 wherein receiving brush data comprises receiving data corresponding to an image.

11. The method of claim 10 further comprising, receiving  
20 markup corresponding to an image effect to apply to the image.

12. The method of claim 1 further comprising, receiving markup corresponding to pen data that defines an outline of a shape.

25

13. The method of claim 1 wherein the markup corresponds to a shape class comprising at least one of the set containing rectangle, polyline, polygon, path, line and ellipse shapes.

5        14. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a geometry-related function to represent a rectangle in the scene graph data structure.

10       15. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a geometry-related function to represent a path in the scene graph data structure.

15       16. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a geometry-related function to represent a line in the scene graph data structure.

20       17. The method of claim 1 wherein the markup is related to hit-testing a visual in the scene graph data structure.

18. The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a

function related to transforming coordinates of a visual in the scene graph data structure.

19. The method of claim 1 wherein the markup is related  
5 to calculating a bounding box of a visual in the scene graph data structure.

20. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function via a  
10 common interface to a visual object in the scene graph data structure.

21. The method of claim 1 further comprising invoking a visual manager to render a tree of at least one visual object  
15 to a rendering target.

22. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place a container object in the scene graph data structure, the  
20 container object configured to contain at least one visual object.

23. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to  
25 place image data into the scene graph data structure.

24. The method of claim 23 wherein causing data in the scene graph to be modified comprises invoking a function to place an image effect object into the scene graph data structure that is associated with the image data.

25. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place data corresponding to text into the scene graph data structure.

26. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to provide a drawing context in response to the function call.

15

27. The method of claim 26 wherein the function call corresponds to a retained visual, and further comprising, calling back to have the drawing context of the retained visual returned to the scene graph data structure.

20

28. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place a three-dimensional visual into the scene graph data structure.

25

29. The method of claim 28 wherein causing data in the scene graph to be modified comprises mapping a two-dimensional surface onto the three dimensional visual.

5        30. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place animation data into the scene graph data structure.

31. The method of claim 30 further comprising  
10 communicating timeline information corresponding to the animation data to a composition engine.

32. The method of claim 31 wherein the composition engine interpolates graphics data based on the timeline to animate an  
15 output corresponding to an object in the scene graph data structure.

33. The method of claim 32 wherein the composition engine is at a low-level with respect to the scene graph.

20        34. The method of claim 1 wherein causing data in the scene graph to be modified comprises invoking a function to place an object corresponding to audio and/or video data into the scene graph data structure.

25

35. The method of claim 1 wherein causing data in the scene graph to be modified comprises changing a mutable value of an object in the scene graph data structure.

5        36. In a computing environment, a system comprising:  
a markup parser;  
an interface coupling the markup parser to a source of markup;  
and a container for visual information of an object model,  
10 the markup parser converting markup received at the interface to method calls of objects in the object mode to modify data in the container for visual information.

37. The system of claim 26 wherein the markup is  
15 converted to a method call to place a tree of visual objects into the scene graph data structure.

38. The system of claim 37 wherein the markup is  
converted to at least one method call to arrange a tree of  
20 visual objects in the scene graph data structure.

39. The system of claim 37 wherein the markup is  
converted to a method call to place a visual collection object  
into the scene graph data structure.

25

40. The system of claim 26 wherein the markup is converted to a method call to place a visual object into the scene graph data structure.

5        41. The system of claim 40 wherein the markup is converted to a method call to associate a brush with the visual object.

10       42. The system of claim 40 wherein the markup is converted to a method call to associate a geometry with the visual object.

15       43. The system of claim 42 wherein the geometry comprises at least one of a set containing an ellipse geometry, a rectangle geometry, a line geometry and a path geometry.

20       44. The system of claim 40 wherein the markup is converted to a method call to associate a transform with the visual object.

45. The system of claim 44 wherein the transform comprises a rotate transform for changing a perceived angle of the visual object.



46. The system of claim 44 wherein the transform comprises a scale transform for changing a perceived size of the visual object.

5 47. The system of claim 44 wherein the transform comprises a translate transform for changing a perceived position of the visual object.

48. The system of claim 44 wherein the transform  
10 comprises a skew transform for changing a perceived skew of the visual object.

49. The system of claim 44 wherein the markup provides animation information associated with the transform, and  
15 wherein the animation information causes transformation data associated with the transform to change over time thereby animating the transformation of the visual object over time.

50. The system of claim 40 wherein the markup is  
20 converted to a method call to associate a color with the visual object.

51. The system of claim 40 wherein the markup is converted to a method call to associate gradient data with the  
25 visual object.

52. The system of claim 40 wherein the markup is converted to a method call to associate a tile brush with the visual object.

5

53. The system of claim 40 wherein the markup is converted to a method call to associate an image with the visual object.

10 54. The system of claim 40 wherein the markup is converted to a method call to associate a drawing comprising drawing primitives with the visual object.

15 56. The system of claim 40 wherein the markup is converted to a method call to associate audio and/or video media d with the visual object.

20 57. The system of claim 40 wherein the markup is converted to a method call to associate an image effect with the visual object.

58. The system of claim 40 wherein the markup is converted to a method call to associate a pen with the visual object, to describe how a shape is outlined.

25

59. The system of claim 40 wherein the markup is converted to a method call to obtain a drawing context associated with the visual object.

5        60. The system of claim 40 wherein the markup is converted to a method call to associate hit testing data with the visual object.

61. The system of claim 40 wherein the markup is  
10 converted to a method call to associate a rectangle with the visual object.

62. The system of claim 61 wherein the markup includes data describing how a source rectangle should be stretched to  
15 fit a destination rectangle.

65. In a computing environment, a system comprising:  
interface means for receiving function calls comprising markup;

20        parsing means for converting the markup to data corresponding to an object model associated with rendering graphics data; and

rendering means for outputting the graphics data.